STAT 672: Homework 3

Tom Wallace

March 21, 2018

SVD and Ridge Regression

Estimated coefficients in ridge regression are given by:

$$\hat{\boldsymbol{\beta}}_{\text{ridge}} = (\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I})^{-1}\mathbf{X}^T\mathbf{y}$$

We take the singular value decomposition of the feature matrix:

$$= ((\mathbf{U}\mathbf{D}\mathbf{V}^T)^T(\mathbf{U}\mathbf{D}\mathbf{V}^T) + \lambda\mathbf{I})^{-1}(\mathbf{U}\mathbf{D}\mathbf{V}^T)^T\mathbf{y}$$
$$= (\mathbf{V}\mathbf{D}^T\mathbf{U}^T\mathbf{U}\mathbf{D}\mathbf{V}^T + \lambda\mathbf{I})^{-1}\mathbf{V}\mathbf{D}^T\mathbf{U}^T\mathbf{y}$$

$\mathbf{U}$ is orthogonal ($\mathbf{U}^T\mathbf{U} = \mathbf{I}$) and $\mathbf{D}$ is diagonal ($\mathbf{D}^T = \mathbf{D}$):

$$= (\mathbf{V}\mathbf{D}^2\mathbf{V}^T + \lambda\mathbf{I})^{-1}\mathbf{V}\mathbf{D}\mathbf{U}^T\mathbf{y}$$

Because $\mathbf{V}$ is orthonormal, i.e. $\mathbf{V}^T = \mathbf{V}^{-1}$, we substitute in $\mathbf{V}\mathbf{V}^T$ for $\mathbf{I}$:

$$= (\mathbf{V}\mathbf{D}^2\mathbf{V}^T + \lambda\mathbf{V}\mathbf{V}^T)^{-1}\mathbf{V}\mathbf{D}\mathbf{U}^T\mathbf{y}$$

And factor out $\mathbf{V}\mathbf{V}^T$:

$$= \mathbf{V}(\mathbf{D}^2 + \lambda)^{-1}\mathbf{V}^T\mathbf{V}\mathbf{D}\mathbf{U}^T\mathbf{y}$$

And again make use of the fact that $\mathbf{V}$ is orthonormal, this time to cancel some terms:

$$= \mathbf{V}(\mathbf{D}^2 + \lambda)^{-1}\mathbf{D}\mathbf{U}^T\mathbf{y}$$

Since $\mathbf{D}$ is diagonal, we can rewrite the expression involving it and λ:

$$\mathbf{D}_\lambda := (\mathbf{D}^2 + \lambda)^{-1}\mathbf{D}$$

$$= \text{diag}\left(\frac{d_1}{d_1^2 + \lambda} \cdots \frac{d_D}{d_D^2 + \lambda}\right)$$

Thus, computation of estimated coefficients in ridge regression via SVD is given by:

$$\hat{\boldsymbol{\beta}}_{\text{ridge}} = \mathbf{V}\mathbf{D}_\lambda\mathbf{U}^T\mathbf{y}$$

■

Efficiency of Computation

There are an inefficient method and an efficient method of re-calculating ridge regression coefficients for a new regularization parameter λ.

In the inefficient method, we recompute the SVD every time we update λ. SVD has complexity on the order of $O(nd^2)$ (with n corresponding to the number of rows of the feature matrix and d corresponding to the number of columns). We then multiply $\mathbf{V}\mathbf{D}_\lambda\mathbf{U}^T\mathbf{y}$. We exploit the associative property of matrices and proceed right to left for maximum efficiency, i.e. $\mathbf{V}(\mathbf{D}_\lambda(\mathbf{U}^T\mathbf{y}))$. A matrix-matrix product $C = AB$, where $A \in \mathbb{R}^{m\times n}$ and $B \in \mathbb{R}^{n\times p}$, costs $2mnp$ flops. In our case, $\mathbf{U}^T \in \mathbb{R}^{d\times n}$ and $\mathbf{Y} \in \mathbb{R}^{n\times 1}$, and so multiplying the two costs $2 \times n \times d \times 1 = 2nd$ flops and results in a $d \times 1$ vector. Multiplying $\mathbf{D}_\lambda \in \mathbb{R}^{d\times d}$ and this $d \times 1$ vector costs d flops (assuming we take advantage of the diagonal structure of $\mathbf{D}_\lambda$) and results in a $d \times 1$ vector. Multiplying $\mathbf{V} \in \mathbb{R}^{d\times d}$ by this $d \times 1$ vector costs $2d^2$ flops. Adding together all these steps, we have $nd^2 + 2nd + d + 2d^2$ flops. Dropping all constant coefficients and only considering the highest-order polynomial, we conclude that the inefficient method costs $O(nd^2)$ flops.

A more efficient method notes that $\mathbf{D}_\lambda$ is the only part of $\mathbf{V}\mathbf{D}_\lambda\mathbf{U}^T\mathbf{y}$ that depends on λ and so we do not need to recompute the SVD for every new value of λ. Assume that we have pre-calculated and cached $\mathbf{U}^T\mathbf{y}$ and $\mathbf{V}$. Modifying the d non-zero values of $\mathbf{D}_\lambda$ to reflect our new value of λ costs $2d$ flops (d flops for addition of the new λ in the denominator, and d flops to divide the numerator by the new denominator). Computing new regression coefficients requires multiplying the new $\mathbf{D}_\lambda$ and $\mathbf{U}^T\mathbf{y}$, which costs $2d^2$ flops, and then multiplying that result by $\mathbf{V}$, which also costs $2d^2$ flops. Adding together these two steps, we have $2d + 2d^2 + 2d^2$ flops. Dropping all constant coefficients and only considering the highest-order polynomial, we conclude that the efficient method costs $O(d^2)$ flops.

In practical terms, since we are fitting regression coefficients on a training dataset with $n = 6000$, the efficient method is 6,000 times faster.

Results

Below, I compare the performance of ridge and lasso regression on the homework dataset for different values of λ. For ridge regression, a regularization parameter value of 0.00390625 achieves the lowest test error (2.81855); for lasso regression, a regularization parameter of $4.46e^{-0.6}$ (i.e. the tested value closest to 0) achieves the lowest test error (2.85948). The ridge method achieves the better minimum test error.

In the below graph, I have omitted some tested values of λ to make things more visually clear.